




ERJU SYSTEM PILLAR

Systems Engineering Management Plan - Annex 04 Architecture Definition Process



Systems Engineering Management Plan - Annex 04 Architecture Definition Process

Author(s)	Jorge Block , Dennis Kunz
Abstract	This document describes all process tasks of the creation of a logical architecture according to ARCADIA.
Config Item	System Engineering Management Plan
Document ID	SEMP Annexes/SEMP process 04-Architecture#722539  Systems Engineering Management Plan - Annex 04 Architecture Definition Process
Classification	Public
Status	Released
Version	2.0
Revision	722539
Last Change Date	02.10.2025
Copyright	Brussels: Europe's Rail Joint Undertaking, 2025

© Europe's Rail Joint Undertaking, 2025

This document is drafted by and belongs to EU Rail.

EU Rail encourages the distribution and re-use of this document, the technical specifications and the information it contains. EU Rail holds several intellectual property rights, such as copyright and trade mark rights, which need to be considered when this document is used.

EU Rail authorises you to re-publish, re-use, copy and store this document without changing it, provided that you indicate its source and include the following: EU Rail trade mark, title of the document, year of publication, version of document.

EU Rail makes no representation or warranty as to the accuracy or completeness of the information contained within these documents. EU Rail shall have no liability to any party as a result of the use of the information contained herein. EU Rail will have no liability whatsoever for any indirect or consequential loss or damage, and any such liability is expressly excluded.

You may study, research, implement, adapt, improve and otherwise use the information, the content and the models in the this document for your own purposes. If you decide to publish or disclose any adapted, modified or improved version of this document, any amended implementation or derivative work, then you must indicate that you have modified this document, with a reference to the document name and the terms of use of this document. You may not use EU Rail's trade marks or name in any way that may state or suggest, directly or indirectly, that EU Rail is the author of your adaptations.

EU Rail cannot be held responsible for your product, even if you have used this document and its content. It is your responsibility to verify the quality, completeness and the accuracy of the information you use, for your own purposes.

Document History


1.1 07.01.2023	Schmidt Steffen (I-NAT-SIBS-ISST)	Reviewed version including Findings from Review X.X
1.2 25.03.2024	SANGO Marc (SNCF / DIR TECHNOLOGIES INNOVATION ET PROJETS GROUPE / IR DIR RECHERCHE - PSF)	Reviewed version including Findings from Review 1.1
1.3 25.08.2025	Jorge Block	Reviewed version including Findings from Review 1.2
2.0 18.09.2025	Dennis Kunz	Approved version based on Review 2.1

2.1 18.09.2025 Dennis Kunz

2.0 18.09.2025 Dennis Kunz

Reviewed version including
Change Request
Approved version based on
Review 2.1

Approval by reviewers (captured at end of 'In Review by System Pillar')

Type of Approval	 Document Review
Approvals	ANTOONS Gilles : Approved , Renard, Marie Pierre (SMO RI MT FR ADC TGMTR3) : Approved , EDDOUS Sayfeddine (SNCF RESEAU / Directions Techniques Réseau / DGII DTR GE SF Solutions) : Approved , Dennis Kunz : Approved
Attachments	Systems_Engineering_Management_Plan_-_Annex_04_Architecture_Definition_Process-comments.xlsx

Approval by approvers (captured at end of 'In Approval by System Pillar')


Type of Approval	 Document Approval
Approvals	SANGO Marc (SNCF / DIR TECHNOLOGIES INNOVATION ET PROJETS GROUPE / IR DIR RECHERCHE - PSF) : Approved , KEFALAS, Georgios : Approved , Schmidt Steffen (I-NAT-GST-ERTM) : Approved , SCHWAN Nico : Approved , Dennis Kunz : Waiting

Table of contents

1 Preamble	6
1.1 Purpose	6
1.2 Intended Audience	6
1.3 Document Context	6
1.4 Glossary	6
2 Content	7
2.1 P4.1 Define architecture and allocate functionalities	7
2.2 P4.2 Refine behaviour	10
2.3 P4.3 Consolidate architecture description	11
2.4 P4.4 Provide input for risk and hazard analysis	13
2.5 P4.5 Apportion requirements	13
3 Appendix	14
3.1 Standards and References	14

1 Preamble

1.1 Purpose

Purpose of the architecture definition process

The Architecture Definition process interacts iterative with the requirements definition process to understand the problem and identify the most satisfactory architecture solution.

The results of the Architecture Definition process are widely used across the life cycle processes. As the process unfolds, insights are gained into the relation between the requirements specified for the system of interest and the emergent properties and behaviours of the system of interest that arise from the interactions and relations between the subsystems.

A Architecture definition may be applied at many levels of abstraction, highlighting the relevant detail that is necessary for the decisions at that level. [SPPR-11500]

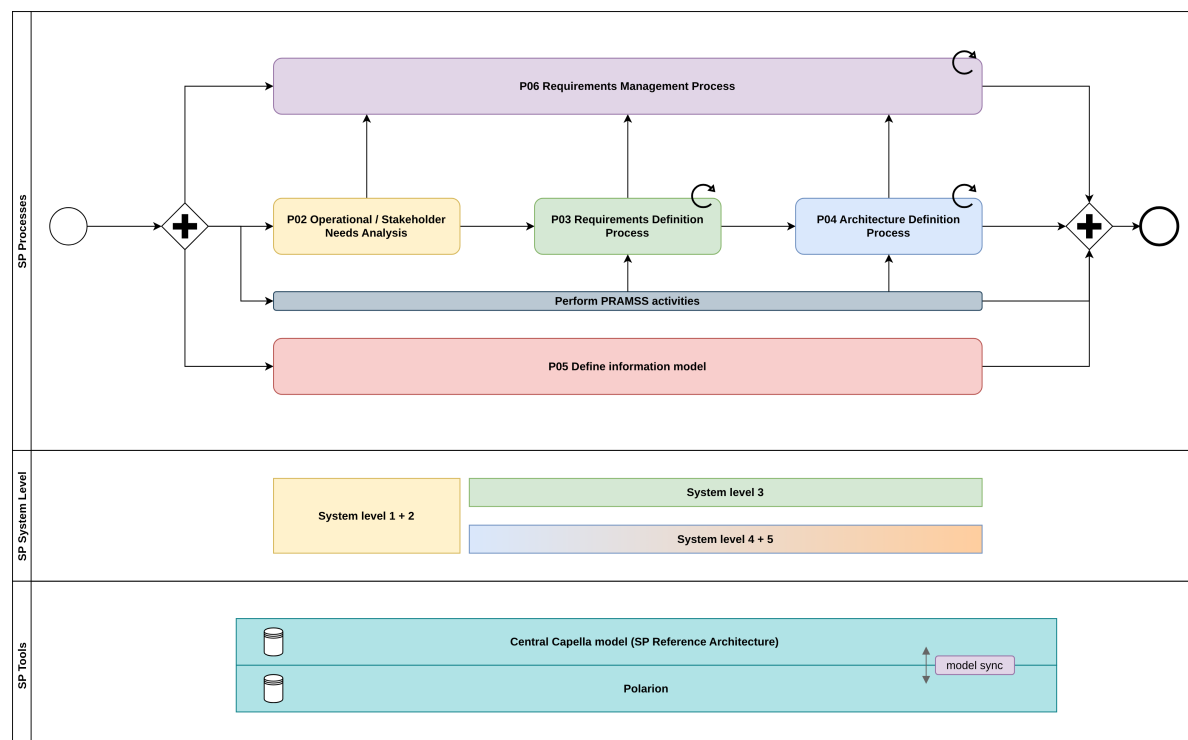
1.2 Intended Audience

The content is valid for all the System Pillar tasks and domains.

1.3 Document Context


This document describes the Architecture Process as part of the System Pillar processes.

System Pillar process overview



[SPPR-4083]

1.4 Glossary


This document primarily outlines the process, while the definitions can be found in  Systems Engineering Management Plan - Annex M2 Viewpoint Guidelines.

2 Content


P04 - Architecture definition process

The development of the system architecture is a preparation step for the subsystem requirements. This is done from a technology-independent and solution-free perspective, it ensures design solution flexibility and adaptability by existing independently of and without imposing design decisions.

The main goal of the architecture is to refine the functions from the system definition into functions allocated to subsystems. Besides, the system of interest is no longer treated as a black-box: Functions are grouped into internal systems, forming the structural basis of the system architecture. Interfaces between internal systems are defined, based on the functional exchanges previously identified. The internal systems are still treated as a black-box.

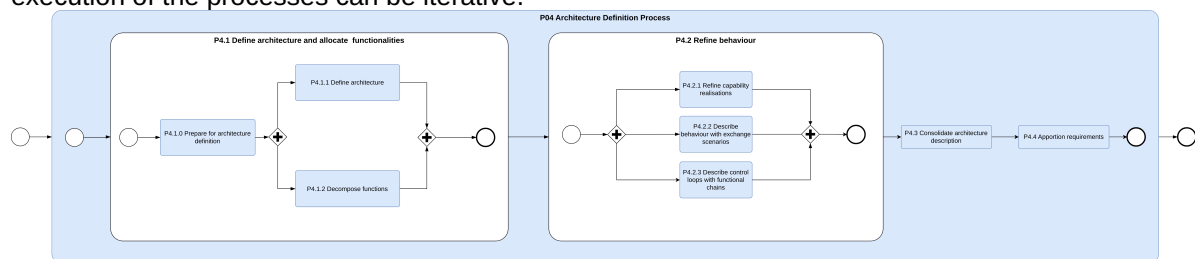
This process contains all tasks to develop the system architecture the tasks can be followed to create content in the **logical architecture** or **physical architecture** using architectural models which are based on ARCADIA and corresponding  SPPR-9823 - Capella2Polarion work items.

- SPPROCESS/10 SEMP V 01_01/SEMP Annex M1 Capella element rules : 722539
- SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722539

Inputs	<ul style="list-style-type: none"> - System definition of the considered system stored in the System Pillar reference architecture model - System requirements specification which contains all required work items based on the document template
Outputs	<ul style="list-style-type: none"> - Architecture of the considered system stored in the System Pillar reference architecture model - System architecture description which contains all required work items based on  Template - System Architecture Description
ID	SPPR-2406

Architecture Definition Process Overview

The following diagram shows the processes and tasks for the definition of the architecture. The execution of the processes can be iterative.



[SPPR-11501]


2.1 P4.1 Define architecture and allocate functionalities

P4.1.0 - Prepare for architecture definition




Description:

This task describes the steps to prepare for the architecture definition.

Steps:

1. **Read** the  Systems Engineering Management Plan - Annex M2 Viewpoint Guidelines to understand the different viewpoints for the architecture definition.
2. **Review and analyse relevant information** and related perspectives that will guide the development of the system architecture. This includes PRAMSS aspects as well.


Note: This information is intended to help build an understanding of the environment for which the solution is needed in order to establish better insight into the stakeholder concern.

3. **Identify key milestones and decisions** in order to define the architecture strategy based on  *ARC-D2.5 Architecture Guidelines* and  *System Concept_CCS - Granularity Concepts and Principles - Main*.
4. **Decide** if there are necessary **enabling systems and/or services** required to support the architecture definition.
5. **Analyse and review the system context** defined in the system definition on higher architecture layer.
6. (Optional) **Check model content** from system definition and perform automatic transfer if possible
 - a. For each reviewed capability that has been evaluated on the higher level:
 - i. **Perform** an automatic transition of relevant **capabilities** in order to transfer the involved functions. It is recommended to use the "Functional Transition" of the selected capability.
 - ii. **Perform** an automatic transition of all the **actors and interfaces** in order to keep the function allocations.
 - iii. **Create** an initial  SPPR-6633 - Functions involvement. under the considered **capability** in order to prepare the viewpoints needed for the system architecture. These capability realisations can be refined in later tasks of this process.
 - iv. **Perform** an automatic transition of the **functions**.

Inputs	<ul style="list-style-type: none"> - System Definition - System Requirements Specification - Stakeholder Requirements Specification - Constraints record - Model elements to be transferred to the next architecture layer
Outputs	<ul style="list-style-type: none"> - Functions involvement - Actors - Interfaces
ID	SPPR-7049

P4.1.1 - Define architecture



Description:







This task serves as a preliminary exploration of the architecture. Its purpose is to broaden the scope of potential solutions through brainstorming sessions, aiming to define one or more alternative solutions. A set of system interfaces between the internal systems of the architecture from the overall system of interest is also defined. This task should be done in parallel with  SPPR-7050 - Decompose functions.

Methods:

- SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722539
- SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722539




Steps:

1. **Define a first set of system/subsystem candidates**, by generating a  SPPR-8378 - Architecture description view based on  SPPR-11000 - Construction method system architecture description. The highest priority requirements and architecture considerations needs to be addressed. Additional candidates may be defined already by system constraints.
 - a. Note: This task includes scanning for relevant technologies, problem patterns, solution patterns, naturally occurring solutions, enhancements to existing systems, heuristics, tactics, determination of architectural entities, discussion with experts.
2. **Rationalise** architecture decisions and define addition alternatives if needed
3. **Perform a trade-off assessment** regarding the architecture alternatives. Take the system constraints into account which may restrict the allocation of system functions into internal systems.

4. **Define selected** system architecture. Record this in an updated version of  SPPR-8378 - Architecture description
5. **Allocate the functions** to in the  SPPR-6628 - Function allocation view by following  SPPR-11003 - Construction method function allocation to develop the system architecture of the system of interest. If a function can not be allocated to only one system the function needs to be split based on  SPPR-7050 - Decompose functions.
6. **Consolidate** system constraints that are relevant for the selected alternative. Keep track of the disregarded constraints that may be not longer applicable.
 - a. Note: Constraints should only be discarded after the corresponding discussion with involved stakeholders, in order to avoid rejected validations in later phases.
7. **Create the architecture trade-off record** to keep track of the reasoning and relevant information of the decision.
8. **Identify system functions** which are not allocated to any system component. If there are such cases, then either delete or merge them with other system functions (if suitable), so that there are not any non-allocated system functions left.
9. **Define internal interfaces** in  SPPR-6626 - Interface description by following  SPPR-10983 - Construction method interface description in the model. This can be interfaces without available constraints or constraint interfaces.


Notes:

- The candidate systems names should be as brief as possible to be manoeuvrable and easy to understand, based on terminology known for stakeholders, or coming from the inputs to the project.
- If there is an information source of internal systems already available (e.g. from collaboration projects), reuse is recommended.

Inputs	<ul style="list-style-type: none"> - System Definition - System Requirements Specification - Stakeholder Requirements Specification - Constraints record - Defined functions - Concepts and other information from railway and PRAMS experts
Outputs	<ul style="list-style-type: none"> -  SPPR-8378 - Architecture description -  SPPR-6628 - Function allocation -  SPPR-6626 - Interface description
ID	SPPR-7097

P4.1.2 - Decompose functions



Description:

This task determines the functions (e.g. through split, breakdown, decomposition) in order to satisfy system requirements and realise system functions. Transited functions may be broken down into smaller, leaf functions to better align them with a sensible allocation among the system elements. This task should be done in parallel with  SPPR-7097 - Define architecture.

Methods:

- SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722539
- SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722539



Steps:

1. **Identify the restrictions** that may influence the splitting of the system functions.
2. **Create the functions**, by redefining the system functions according to the required level of detail on each  SPPR-6633 - Functions involvement.
3. **Create the system functional exchanges**: Redefine the functional exchanges according to the defined exchange principle on each  SPPR-6633 - Functions involvement.
4. **Check** that all **realisation links** from the **system functional exchanges to the corresponding system functional exchanges** and from the **system functions to the corresponding system**


functions defined in the system analysis are set correctly - if this relationship is missing, then **create** it.

- a. **Review** the completeness of **in relation to** in order to verify that all the functional exchanges have allocated exchange items.

5. If the functional exchanges do not have allocated exchange items:

- a. **Refine** the **data model**, in order to create additional data elements.
- b. **Allocate** new **exchange item** to the functional exchanges in the corresponding
 SPPR-6633 - Functions involvement based on  SPPR-10055 - Construction method exchange items.

6. Remove the transferred parent system function from the system architecture viewpoints.

Inputs	- Functions - System Requirements Specification
Outputs	-  SPPR-6633 - Functions involvement
ID	SPPR-7050

2.2 P4.2 Refine behaviour

P4.2.1 - Refine capability realisations



Description:



This task captures the refinements or adjustments of the system capabilities. For the most part, the system capabilities/systems capability realisations can stay at the same level of granularity for the system architecture. **The amount of refinement of capability realisations at system architecture level should be kept to the minimum necessary.**

Methods:

- SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722539
- SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722539

Steps:

1. If the boundaries of the capability realisation need to be redefined due to the split of system functions into system functions, **define a procedure to decompose (or tailor) the capabilities** appropriately and the system functions defined in  SPPR-6633 - Functions involvement. Some system constraints may restrict the splitting of capabilities.
2. **Consolidate the system capability realisations** in the  SPPR-6629 - Capabilities definition in order to show their relations with the system actors. Some system constraints may restrict the involvement of actors in the realised capabilities.
3. **Review the pre- and postconditions** of each capability realisation in order to refine them if needed
4. **Review the traceability** with the system capabilities and create relationships if it is missing in order to ensure consistency.

Inputs	- System capabilities (transferred) -  SPPR-6633 - Functions involvement
Outputs	-  SPPR-6629 - Capabilities definition
ID	SPPR-7094

P4.2.2 - Describe behaviour with exchange scenarios

Description:

This task refines the scenarios which were created during system definition. The lifelines are updated with the creation of the internal systems.



Methods:




- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722539*
- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722539*

Steps:

a. **Define pre- and postcondition** see

SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722539 for more information.

b. Follow  SPPR-7249 - Construction method exchange scenario to create  SPPR-8176 - Exchange scenario diagram.

Inputs	-  SPPR-8176 - Exchange scenario (of System Definition) -  SPPR-6633 - Functions involvement
Outputs	-  SPPR-8176 - Exchange scenario (of Architecture Description)
ID	SPPR-8404





P4.2.3 - (Optional) Describe control loops with functional chainsDescription:

This task refines the functional chains of the capability realisations with the decomposed system functions.

Methods:




- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722539*
- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722539*

Steps:

1. Follow  SPPR-11021 - Construction method functional chain to create  SPPR-8178 - Functional chain description diagram.
2. Follow  SPPR-11315 - Construction method control loop to create  SPPR-11362 - Control loop diagram.

Notes:

- If it is necessary, functional chains can be assembled and reused in order to express more complex scenarios.
- If there are already functions that exist in the model, the sequential order of these functions can be described in a function chain. The functional chain should be located underneath an assumed corresponding capability realisation.

Inputs	-  SPPR-6633 - Functions involvement
Outputs	-  SPPR-11362 - Control loop -  SPPR-8178 - Functional chain description
ID	SPPR-7095

2.3 P4.3 Consolidate architecture description**P4.3 - Consolidate architecture description**Description:


This task consolidates the functional flows to maximise the reuse of system functions and their functional exchanges in more than one capability realisation and ensure that the system functions instead of the system functions are used. Furthermore this task ensures that there are no gaps or duplicates between defined internal systems or interfaces. In addition traceability between elements of the system architecture and system definition will be established. Finally system architecture description document is created.

Methods:



- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M ARCADIA Capella Modelling Rules : 722539*
- *SPPROCESS/10 SEMP V 01_01/SEMP Annex M2 Capella diagram rules : 722539*

Steps:




Consolidate functionality to maximise the reuse of functions and their functional exchanges.

1. **Perform** an analysis of **duplicated functions** in  **SPPR-9580 - Functional flow consolidation diagram**, regarding their name, allocation and realised links.
2. **Merge functions** by re-linking/re-allocating the changed functions in the Consolidated functional flow definition in order to disconnect the duplicated functions from the rest of the model - only if the commonalities are clear and if they are identical or can be made identical by redesigning them.
3. **Delete** involvements of the duplicated functions which will be removed in the corresponding functional chain definition and exchange scenario definition.
4. **Replace** the **duplicated functions** with the correct merged functions in the corresponding functional chains, exchange scenarios, functional flows.
5. **Remove duplicates** from the database, if they have been fully disconnected from the rest of the model (no more involvement in capabilities, scenarios, or functional chains and no functional exchanges).
6. **Review** that the **functions** displayed in the Consolidated functional flow definition are all the functions defined in the database, i.e. all **functions** are connected to each other and there are no unused elements left (e.g. functional exchanges without exchange items).



Consolidate internal systems

1. **Review the systems** by checking their description defined in the  **SPPR-8378 - Architecture description**.
2. **Consolidate names and rationales** in the  **SPPR-8378 - Architecture description** to ensure that their purpose can be derived from that name and the rationales.
3. **Identify duplication's** thanks to the name and the rationales, choose one and remove the other.
4. **Identify missing systems** if there are any, go back to the previous task Define system architecture to validate if this is really the case.

Consolidate exchange items and data of interfaces to maximise the reuse of exchange items and create new ones if necessary.

1. **Review** completeness and consistency of  **SPPR-8177 - Exchange items description** in relation to functions and functional exchanges using  **SPPR-6633 - Functions involvement**.
2. If the functional exchanges do not have allocated exchange items, **refine the data model**.
3. **Allocate** new **exchange items** to the functional exchanges in the corresponding  **SPPR-6633 - Functions involvement**.
4. **Report traceability** (optional) between exchange items and functional exchanges in a traceability matrix in order to verify that all exchange items are allocated to functional exchanges.

Consolidate traceability


1. **Review the relationships** between model elements either  **SPPR-10247 - How to establish traceability in between System Analysis and Logical Architecture model elements in Capella** OR  **SPPR-10091 - How to establish traceability in between Logical Architecture and Physical Architecture model elements in Capella**.
2. **Review the correctness** of the relationship, if the relationship exists.
3. **Create a relationship** between the element and a corresponding model element, if the relations does not exist.













Perform review

1. **Transfer content** to Polarion using the Capella2Polarion bridge.

2. **Generate architecture description** based on  Template - System Architecture Description
3. **Perform quality review** with EET.
4. **Perform SPPROCESS/SEMP Annex D Processes/Work product review process** : 722539.

Notes:

- If no realisation links can be generated automatically for a given model element, establish a new generic trace relationship between them.
- Further guidance can be found in  Template - System Architecture Description

Inputs	<ul style="list-style-type: none"> -  SPPR-6633 - Functions involvement -  SPPR-11362 - Control loop -  SPPR-8178 - Functional chain description -  SPPR-8378 - Architecture description -  SPPR-6628 - Function allocation -  SPPR-8176 - Exchange scenario
Outputs	<ul style="list-style-type: none"> -  SPPR-6633 - Functions involvement (updated) -  SPPR-11362 - Control loop (updated) -  SPPR-8178 - Functional chain description (updated) -  SPPR-8378 - Architecture description (updated) -  SPPR-6628 - Function allocation (updated) -  SPPR-8176 - Exchange scenario (updated)
ID	SPPR-7101

2.4 P4.4 Provide input for risk and hazard analysis

P4.4 - Provide input for risk and hazard analysis

Description:

This task conducts the risk and hazard analysis for the system architecture. Together with the PRAMSS team, risks of the system will need to handle will be identified and developed. The analysis includes all types of hazards and risks (PRAMSS, economic risks, market risks, risks for the stakeholder like migration risks).

Steps:

1. **Transfer content to Polarion** using the Capella2Polarion bridge.
2. The tasks of risks and hazards related to PRAMS (Performance, Reliability, Availability, Maintainability) is generally defined by the *SPPRAMS/Phase 3/ERJU PRAMS Plan : 722539* this needs to be followed.


Inputs	- System Architecture
Outputs	<ul style="list-style-type: none"> - Hazards - Risks - System requirements - Application conditions
ID	SPPR-2425

2.5 P4.5 Apportion requirements

P4.5 - Apportion requirements

Description:

This task will be defined at a later stage.

In general, the  Systems Engineering Management Plan - Annex 03 Requirements Definition Process can be performed again after creation of the architecture for each internal system.

Inputs	- System Requirements Specification - System Architecture Description
Outputs	- System PRAMS Apportionment Report - System Security Apportionment Report
ID	SPPR-7133

3 Appendix

3.1 Standards and References

There are currently no specific references used by this document.